

The percolation critical polynomial as a graph invariant

Christian R. Scullard*

Lawrence Livermore National Laboratory, Livermore CA 94550, USA

(Dated: November 3, 2011)

Every lattice for which the bond percolation critical probability can be found exactly possesses a critical polynomial, with the root in $[0, 1]$ providing the threshold. Recent work has demonstrated that critical polynomials can be defined on any periodic lattice. In general, the polynomial depends on the lattice and on its decomposition into identical finite subgraphs, but once these are specified, the polynomial is essentially unique. On lattices for which the exact percolation threshold is unknown, the polynomials provide approximations for the critical probability with the estimates appearing to converge to the exact answer with increasing subgraph size. In this paper, I show how the critical polynomial can be viewed as a graph invariant like the chromatic and Tutte polynomials. Like these, the critical polynomial is computed on a finite graph and may be found using the deletion-contraction algorithm. This allows efficient calculation on a computer, and I present such results for the kagome lattice using subgraphs of up to 36 bonds. For one of these, I find the prediction $p_c = 0.52440572\dots$, which differs from the numerical value, $p_c = 0.52440503(5)$, by only 6.9×10^{-7} .

Percolation is the study of the formation of random clusters on lattices. Given an infinite lattice, we declare each bond to be open with probability p , and closed with probability $1 - p$. The resulting random clusters grow in average size with p until we reach the critical point, p_c , at which an infinite cluster first appears. The determination of p_c is an unsolved problem in all but one dimension and on two-dimensional lattices that are self-dual 3-uniform hypergraphs [1]. An example is shown in Figure 1a, where the shaded triangle can represent any configuration of bonds or sites. Critical thresholds on these lattices are given by the Ziff criterion [2],

$$P(A, B, C) = P(\bar{A}, \bar{B}, \bar{C}) \quad (1)$$

where $P(A, B, C)$ is the probability that all three vertices are connected through open paths, and $P(\bar{A}, \bar{B}, \bar{C})$ is the probability that none are connected. Application of (1) to find a bond threshold results in a polynomial in p , with order at most equal to the number of bonds in the unit triangle. We may consider each bond of an n -bond triangle to have a different probability, resulting in a critical surface,

$$f(p_1, \dots, p_n) = 0 \quad (2)$$

where the function f is at most first order in any of its arguments. I refer to this property as “linearity”. Examples of such critical surfaces include the square lattice,

$$S(p_1, p_2) = 1 - p_1 - p_2, \quad (3)$$

and the honeycomb lattice,

$$H(p_1, p_2, p_3) = 1 + p_1 p_2 p_3 - p_1 p_2 - p_1 p_3 - p_2 p_3. \quad (4)$$

Note that critical surfaces may be multiplied by an arbitrary constant, or even another function with no root in $[0, 1]$, without changing the predicted bond threshold. Removing the former freedom by requiring the constant term to be +1, and the latter by demanding linearity, makes the critical polynomial unique. Consider now

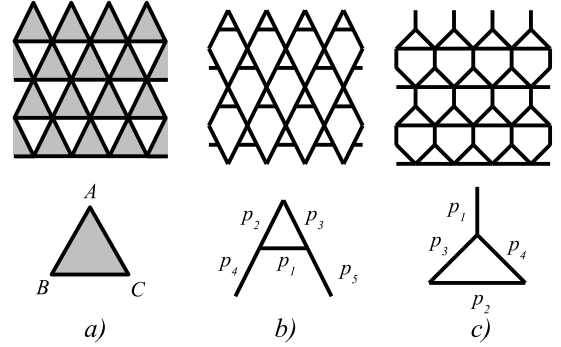


FIG. 1. A 3-uniform hypergraph. The shaded region need not be a simple triangle.

the martini-A lattice, with the probability assignments shown in Figure 1a. If we set $p_1 = 1$, this bond is contracted and its end vertices merged, the result being the honeycomb lattice with two bonds doubled in parallel. This doubled bond can be replaced by a single effective bond, so we get,

$$H(1 - [1 - p_2][1 - p_3], p_4, p_5). \quad (5)$$

By setting $p_1 = 0$, we delete the bond, and the result is the square lattice with bonds doubled in series. Thus, we have,

$$S(p_2 p_4, p_3 p_5). \quad (6)$$

The A lattice is a 3-uniform hypergraph and its critical surface given by (1). The only way to reduce to the correct deleted and contracted surfaces and preserve the required linearity property is to set

$$A(p_1, p_2, p_3, p_4, p_5) = p_1 H(1 - [1 - p_2][1 - p_3], p_4, p_5) + (1 - p_1) S(p_2 p_4, p_3 p_5), \quad (7)$$

which has the formal appearance of an average of the two special cases. This form is general for lattices that can be solved using (1), and constitutes the deletion-contraction formula for critical surfaces of such systems. As another example, we find for the martini-B lattice (Figure 1b),

$$B(p_1, p_2, p_3, p_4) = p_4 S(p_1, 1 - [1 - p_3][1 - p_4]) + (1 - p_4) S(p_1 p_3, p_4). \quad (8)$$

The deletion-contraction algorithm can be used to extend the definition of the critical polynomial to lattices for which the critical threshold is not known exactly. For the kagome lattice of Figure 2a, we may write,

$$K(p_1, p_2, p_3, p_4, p_5, p_6) = p_4 B(1 - [1 - p_5][1 - p_6], p_1, p_2, p_3) + (1 - p_4) A(p_1, p_2, p_3, p_5, p_6), \quad (9)$$

where A and B are given by (7) and (8). Critical polynomials are found by setting all probabilities equal, and in this case $K(p, p, p, p, p, p) = 0$, i.e.,

$$1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6 = 0 \quad (10)$$

which gives the well-known [3, 4] approximation $p_c = 0.52442971\dots$, compared to the numerical [5] $p_c = 0.52440502(5)$. Thus, a critical polynomial may be defined on any lattice for which known lattices appear upon deletion and contraction of a bond, but there is no guarantee that the resulting prediction will be exact. In fact, the definition extends to any lattice, for if either of the lattices appearing in the deletion-contraction formula has an unknown threshold, one simply applies the formula recursively and continues this procedure until known cases appear. This method is used to compute other graph invariants such as the chromatic and Tutte polynomials [6, 7].

In the following, I call the subgraph of a lattice, on which different probabilities are assigned, the “base” of the process, which is then tiled to form the infinite lattice. For example, in Figures 2a, 2b, 3, 4 are bases for the kagome lattice consisting of one, two, four, and six unit cells. Some known and conjectured properties of critical polynomials are as follows:

1. They are unique (up to a trivial multiplication by a constant) once the base and tiling are specified [8, 9]. In particular, they are independent of the bonds chosen to delete and contract at each step. In this sense, the critical polynomial is a property of a finite graph.
2. If the single-cell prediction is not exact, it is generally very close, usually within 10^{-5} of the numerically determined threshold [10]. In addition, proper choices of larger bases lead to predictions closer to the exact answer, with accuracy increasing with base size. This is not proven, but the argument for it is the wide variety of examples in which it is clearly seen to be true.

3. Conversely, if the critical polynomial gives the exact threshold for a base of a single unit cell, as is the case for self-dual 3-uniform hypergraphs, then any critical polynomial using a larger base makes the same prediction, i.e. the original polynomial always factors out. This is also a conjectured property but it would seem to be necessary for consistency, and I have found no counter-examples.

4. In many cases, if the single-cell polynomial does not give the exact answer, then it can be shown [8] that no critical polynomial derived from a finite-sized base will give the correct percolation threshold. This is true for the kagome lattice, as discussed below.

Critical polynomials have been found for all the Archimedean lattices [8, 10, 11]. The polynomial (10) was originally found by Wu using his “homogeneity” [3] assumption and recently, he extended the method to include kagome subnets [9], with excellent results [12]. In fact, the homogeneity approximation gives a prediction for the full q -state Potts critical frontier, but at present it appears limited to the kagome lattice and its subnets as it relies on a transformation from the triangular lattice to these kagome-type graphs. Although I focus on the kagome lattice in this paper, mostly because of the interest it has attracted over the years (e.g. [3, 13, 14]), the method presented here may be applied on any periodic lattice for either site or bond percolation.

In previous work, critical polynomials were found “by hand”, with the (4,6,12) lattice and its 18-bond unit cell probably representing the limit of what one would be inclined to do this way (see the appendix of [8]). However, the recursive nature of the algorithm makes it an ideal problem for a computer, and in this paper, I report the results of using a program to calculate critical polynomials on the kagome lattice for bases containing 12, 24, and 36 bonds. At each step, the program chooses a bond, and finds the two graphs that result from its deletion and contraction. It knows a small number of graphs (e.g., triangular, honeycomb, and square lattices), and it repeats the deletion-contraction algorithm recursively until it recognizes all the lattices it has found. The output is a set of function definitions, like equations (7), (8) and (9) which can be evaluated in a computer algebra package to get the critical surface and then the critical polynomial. Of course, there are many issues to overcome in programming this scheme, and a more full account will be given elsewhere. Here, I show some examples of its operation on the kagome lattice.

Base of 2 unit cells

The first extension beyond a single unit cell base is that shown in Figure 2, in which we employ a base using

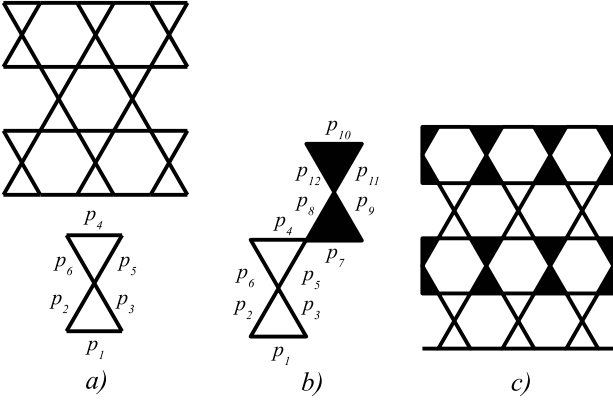


FIG. 2. a) a base for the kagome lattice using two unit cells, shown in two different shades. Each bond on the base has a different probability; b) a tiling for a); another tiling that is equivalent to b).

two unit cells which are indicated by different colors in Figure 2a. The polynomial can be written in the factored form

$$-(1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6) \times (-1 - p^2 - 2p^3 + 10p^4 - 10p^5 + 3p^6). \quad (11)$$

We recognize the first term in brackets as the polynomial for the 6-bond base. The second term has no real roots, and thus the prediction here is the same as the one we found previously, $p_c = 0.52442971\dots$. There is no other way to tile this base to give a different result.

Bases of 4 unit cells

Using the base consisting of 4 unit cells shown in Figure 3a, we may tile it in two different ways as indicated in Figures 3b and 3c. Starting with Figure 3b, the critical polynomial can be written in the factored form

$$-(1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6) \times (-1 - p^2 - 2p^3 + 10p^4 - 10p^5 + 3p^6) \times (1 - 2p^2 - 4p^3 + 7p^4 + 24p^5 - 28p^6 - 64p^7 + 172p^8 - 184p^9 + 110p^{10} - 36p^{11} + 5p^{12}) \quad (12)$$

Once again, the first term in brackets is just the 6-bond polynomial (10), and the others have no roots in $[0, 1]$. The prediction is again the same as the 6-bond estimate.

Things finally become more interesting when we use the staggered embedding in Figure 3c. In this case, the polynomial is

$$1 - 6p^4 - 24p^5 - 24p^6 - 24p^7 + 27p^8 + 552p^9 + 1056p^{10} - 1224p^{11} - 8548p^{12} - 4872p^{13} + 68568p^{14} - 50664p^{15} - 226650p^{16} + 643944p^{17} - 843684p^{18} + 684384p^{19} - 368886p^{20} + 133152p^{21}$$

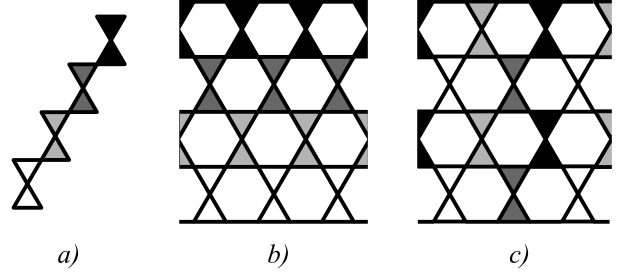


FIG. 3. A 24-bond base for the kagome lattice with two inequivalent embeddings.

$$-31068p^{22} + 4248p^{23} - 259p^{24} = 0, \quad (13)$$

and the solution on $[0, 1]$ is $p_c = 0.52440672\dots$, which differs from the numerical result by 1.7×10^{-6} , a great improvement over the single-cell 6-bond case. There does not appear to be a way to get a better prediction using a 24-bond base.

Bases of 6 unit cells

There are many options for bases of 36 bonds, and I will discuss only three here. If we take the 6-unit cell version of the base shown in Figure 3a with the embedding analogous to Figure 3b, we once again find a polynomial in which (10) appears as a factor. Thus, for this base and embedding, we get the 6-bond estimate again. It is probably safe to assume that this trend continues with larger bases and embeddings of this type.

Turning now to the base shown in Figure 4a, in which the embedding is indicated by the matching shapes on the external vertices, we get the polynomial,

$$1 - 3p^4 - 12p^5 - 20p^6 - 60p^7 - 132p^8 + 56p^9 + 684p^{10} + 1440p^{11} + 2108p^{12} + 2052p^{13} - 10452p^{14} - 68708p^{15} - 82980p^{16} + 280152p^{17} + 1316026p^{18} - 49980p^{19} - 12878976p^{20} + 5124684p^{21} + 90816816p^{22} - 199458252p^{23} - 12979085p^{24} + 816398808p^{25} - 1939348056p^{26} + 2677229528p^{27} - 2575935942p^{28} + 1832168220p^{29} - 984362272p^{30} + 400507236p^{31} - 121897767p^{32} + 26954680p^{33} - 4096134p^{34} + 382956p^{35} - 16617p^{36} = 0, \quad (14)$$

which has solution on $[0, 1]$ $p_c = 0.52440607\dots$, differing from the numerical value by 1.1×10^{-6} . A different base and embedding is shown in Figure 4b and has polynomial,

$$1 - 6p^4 - 24p^5 - 14p^6 + 36p^7 + 39p^8 - 100p^9 - 462p^{10} + 780p^{11} + 4583p^{12} + 4812p^{13} - 9276p^{14} - 71600p^{15} - 85626p^{16} + 312336p^{17} + 1091146p^{18} - 509340p^{19} - 9675936p^{20} + 5297340p^{21} + 66607704p^{22} - 151097304p^{23} - 5319734p^{24} + 610494828p^{25}$$

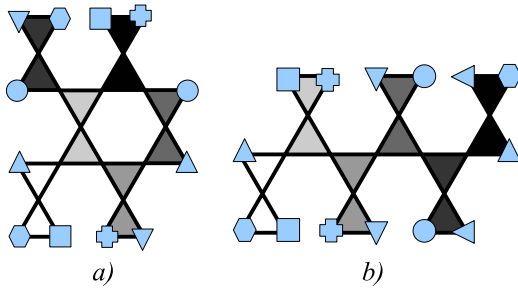


FIG. 4. 36-bond bases for the kagome lattices. Matching shapes on the external vertices indicate how each base is tiled to create the lattice.

$$\begin{aligned}
 & -1461237180p^{26} + 2022998000p^{27} \\
 & -1949295060p^{28} + 1387593528p^{29} \\
 & -745850356p^{30} + 303533928p^{31} \\
 & -92388675p^{32} + 20427736p^{33} - 3103578p^{34} \\
 & + 290052p^{35} - 12579p^{36} = 0,
 \end{aligned} \tag{15}$$

with solution in $[0, 1]$, $p_c = 0.52440572\dots$, slightly better than the previous estimate and within 6.9×10^{-7} of the numerical value.

Clearly, we are observing the predictions converging to the exact answer with increasing base size. From these few examples it also appears that the estimates are converging from above, as they all seem to be greater than the numerical value. However, I know of no argument that guarantees this trend will continue. Note also that no polynomial for the kagome lattice derived from any finite-size base will give the exact answer. This can be seen as follows. Using the base in Figure 4a, we delete many of the bonds to leave a single unit cell and a few connecting bonds, as in Figure 5a. Now, we may contract the connecting bonds to recover the kagome system in which the base is a single cell, as in Figure 5b. By uniqueness, the prediction for this case must be equation (10), which contradicts the prediction found by setting all bonds equal in the full critical surface, i.e. equation (14). Thus, if the single-cell polynomial does not provide the exact threshold, and for the kagome lattice it does not, then, although we can get arbitrarily close by using ever larger bases, no finite critical polynomial derived in this way will ever solve the problem.

I have presented critical polynomials for the kagome lattice, up to bases of 36 bonds. This is the limit of the current implementation of the algorithm, as it becomes progressively more difficult to add bonds due to the exponential complexity. It is not uncommon for a large calculation to produce over a million function definitions. Nevertheless, there is significant room for improvement in the efficiency, as the rule used to choose the bond for deletion and contraction at each step can

have a significant effect on the rate of reduction to the known simple cases. I have hardly explored this issue and presently use what amounts to a random bond se-

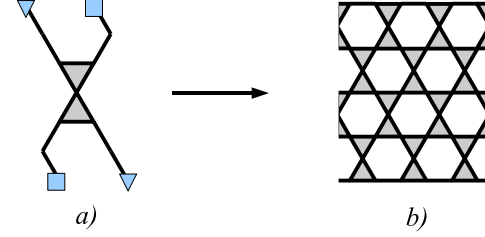


FIG. 5. a) the subgraph and embedding resulting from deleting 22 bonds in Figure 4a, which is a single cell of kagome lattice and some connecting bonds; b) contracting the connecting bonds gives the kagome lattice partitioned into unit cells.

lection, rejecting only those choices that lead to undue complications. Moreover, this algorithm is perfect for a parallel implementation as it would require little inter-processor communication. It remains to be determined how much extra performance can be wrought from these considerations, but the problem will hopefully be seen as an interesting computational challenge.

I thank Robert Ziff for helpful advice on this manuscript. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

* scullard1@llnl.gov

- [1] B. Bollobás and O. Riordan, arXiv:1001.4674 (2010).
- [2] R. M. Ziff, Phys. Rev. E **73**, 016134 (2006).
- [3] F. Y. Wu, J. Phys. C **12** (1979).
- [4] C. R. Scullard and R. M. Ziff, Phys. Rev. E **73**, 045102(R) (2006).
- [5] X. Feng, Y. Deng, and H. W. J. Blöte, Phys. Rev. E **78**, 031136 (2008).
- [6] B. Bollobás, *Modern Graph Theory* (Springer, 1998).
- [7] G. Haggard, D. J. Pearce, and G. Royle, ACM Trans. Math. Softw. **37**, 24:1 (2010).
- [8] C. R. Scullard, J. Stat. Mech., P09022 (2011).
- [9] F. Y. Wu, Phys. Rev. E **81**, 061110 (2010).
- [10] C. R. Scullard and R. M. Ziff, J. Stat. Mech., P03021 (2010).
- [11] C. R. Scullard and R. M. Ziff, Phys. Rev. Lett., 185701 (2008).
- [12] C. Ding, Z. Fu, W. Guo, and F. Y. Wu, Phys. Rev. E **81**, 061111 (2010).
- [13] R. M. Ziff and H. Gu, Phys. Rev. E **79**, 020102 (2009).
- [14] C. Tsallis, J. Phys. C: Solid State Phys. **15**, L757 (1982).